Chapter 5: Function-Induced Persistence
Topics in Computational Topology: An Algorithmic View

Given a domain X, its topology provides a rather coarse description of it. However, functions or maps defined on X give us powerful ways to encode various both geometric and other type of information of X. For example, the Gaussian curvature function $\kappa : X \to \mathbb{R}$ describes the local geometric shape of X. Or imagine that X is a terrain on earth, and $h : X \to \mathbb{R}$ is the elevation of each point; see the right figure for an illustration. Or imagine that X models the hidden space where a social network is sampled from, and a point represents a person. We can use a function $I : X \to \mathbb{R}$ that sends each $p \in X$ to $I(p)$, which is the person p's influence over her/his friends. These two examples are *real-valued functions* defined over X, also called a *scalar function* or a *scalar field*. One can also have more complex maps, such as multivariate functions $f : X \to \mathbb{R}^d$, which encodes multi-dimensional properties of X.

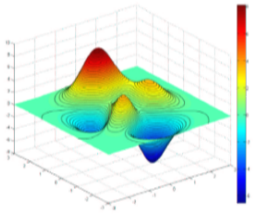


Figure 1: Red color indicates higher elevation.

In this section, we first focus on real-valued functions. In Section 1, we introduce one type of topological information of X w.r.to a scalar field on a smooth function. We then talk about its analog in the discrete setting in Section 1. Next, in Section 3, we introduce an important family of topological description of a domain induced by a function defined on it, called function-induced persistence. Last in Section 5, we give a simple and efficient algorithm for computing the 0-th dimensional function-induced persistence which takes only near linear running time (in constrast the $O(n^3)$ running time in the general case).

1 Morse functions and critical points

1.1 Gradients and critical points

Let us first consider a simple scalar function defined on the real line: $f : \mathbb{R} \to \mathbb{R}$; the graph of such a function is shown in Figure 2 on the right. Recall from calculus the definition of the *derivative* of a function at any point $x \in \mathbb{R}$ is defined as:

$$Df(x) = \frac{d}{dx} f(x) = \lim_{t \to \infty} \frac{f(x+t) - f(x)}{t} \tag{1}$$

Intuitively, $Df(x)$ gives the rate of change of the value of f at x. This can be visualized as the slope of the tangent line of the graph of the function f at x. The critical points of f are the set of points x such that $Df(x) = 0$ – for the function defined on the real line, there are two types of critical points in the generic case: maxima and minima. These points are "critical" because they mark where the behavior of f changes.

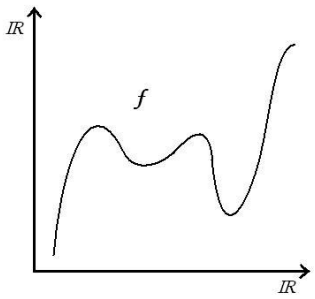


Figure 2: The graph of a function $f : \mathbb{R} \to \mathbb{R}$ defined on the real line

Now suppose we have a real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ defined on $\mathbb{R}^d$. We can draw the graph of this function in $\mathbb{R}^{d+1}$; recall Figure 1 where we show the graph of a function $h : \mathbb{R}^2 \to \mathbb{R}$. Imagine that we are at a point $x \in \mathbb{R}^d$. As we move a little around x, the rate of change of f differs depending on which direction we move. This gives rise to the *directional derivative* $D_v f(x)$ *at* x *in direction (unit vector)* $v \in \mathbb{R}^d$, defined as:

$$D_v f(x) = \lim_{t \to \infty} \frac{f(x + t \cdot v) - f(x)}{t} \tag{2}$$

Definition 1.1 (Gradient and critical points) *Given a differentiable function* $f : \mathbb{R}^d \to \mathbb{R}$ *and* $x \in \mathbb{R}^d$*, the* gradient ∇f *of* f *at* x *is defined as:*

$$\nabla f(x) = [\frac{\partial f}{\partial x_1} \ \frac{\partial f}{\partial x_2} \ \cdots \ \frac{\partial f}{\partial x_d}]^T, \tag{3}$$

where $< x_1, x_2, \ldots, x_d >$ *represents a coordinate system for* $\mathbb{R}^d$*.*

Equivalently, $\nabla f(x)$ *is along the direction* v *where* $D_v f(x)$ *is maximized, and the magnititude* $\|\nabla f(x)\|$ *of* $\nabla f(x)$ *is the value of this maximal directional derivative.*

A point $x \in \mathbb{R}^d$ *is a* critical point *if* $\nabla f(x) = [0 \ 0 \ \cdots \ 0]^T$*. If* x *is critical, then* $f(x)$ *is called a* critical value *for* f*.*

In other words, the gradient of f at x specifies the steepest descendin direction of f with its magnititude being the rate of change along that direction. The critical points of f are those points where the directional derivative vanishes in all directions – locally, the rate of change for f is zero no matter which direction one deviates from x. See Figure 3 for the three types of critical points in a generic setting for a function $f : \mathbb{R}^2 \to \mathbb{R}$: mimina, saddle points, and maxima.

Finally, given a differentiable function $f : M \to \mathbb{R}$ defined on a smooth manifold, we can extend the above definitions of gradients and critical points to it. We will not define it formally here. Intuitively, at a point $x \in M$, we now consider the tangent space of M at x, denoted by TM_x. Roughly speaking, within an infinitesmally small neighborhood of x, one can imagine that the function f is defined on TM_x. If the manifold M is of dimension m, then TM_x is $\mathbb{R}^m$. We can then define the gradient of f with respect to TM_x – that is, the gradient ∇f is a vector field $\nabla f : M \to TM$, and $\nabla f(x) \in TM_x$ represents the steepest descending direction of f among all directions $v \in TM_x$ with its magnititude being the rate of change along this direction. A point x is a *critical point* if its gradient $\nabla f(x)$ vanishes at x.

1.2 Morse functions and Morse Lemma

From the first derivative of a function we can determine critical points. We can say much about the "type" of the critical points by inspecting the second derivatives of f around a point x. Specifically, consider the Hessian Matrix.

Definition 1.2 (Hessian Matrix) *A Hessian Matrix of a second order differentiable function* $f : \mathbb{R}^d \to \mathbb{R}$ *at* x *is the matrix of second derivatives,*

$$Hessian(x) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2} 2(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1}(x) & \frac{\partial^2 f}{\partial x_d \partial x_2} 2(x) & \cdots & \frac{\partial^2 f}{\partial x_d \partial x_d}(x) \end{vmatrix}$$

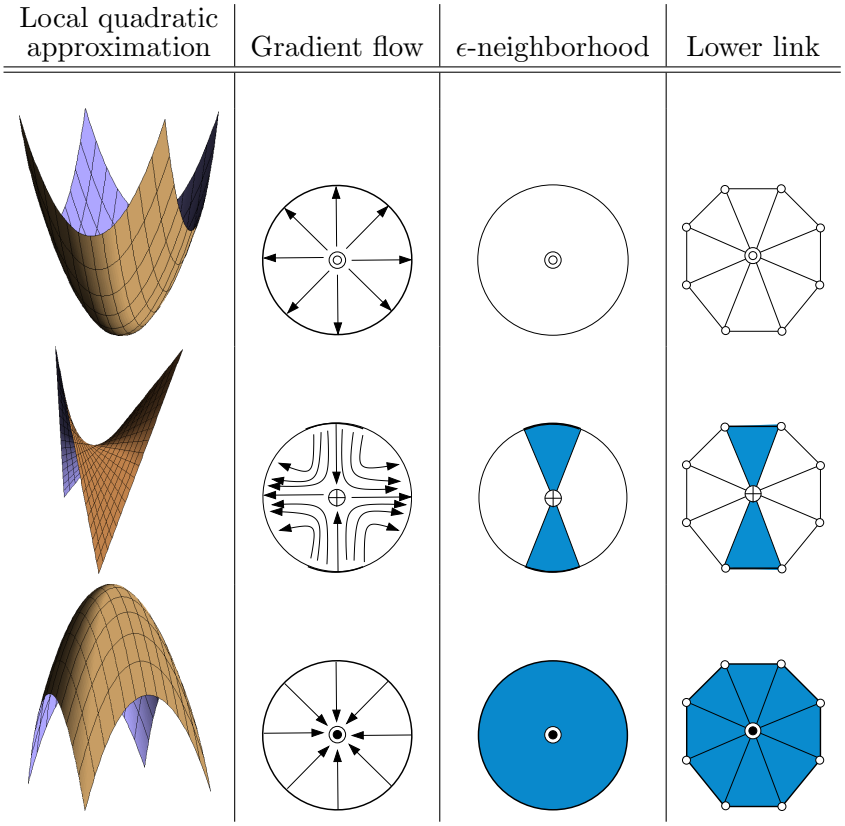


Figure 3: Local neighborhoods of critical points. Top row: minimum; middle row: saddle; bottom row: maximum.

Definition 1.3 (Degenerate critical points) *A critical point x of f is* degenerate *if $Det(Hessian(x)) = 0$ ($Hessian(x)$ is not full rank). Otherwise, the critical point x is considered* non-degenerate.

For example,

$$Let f = x^2 + 1 \tag{4}$$
$$f'(x) = 2x \tag{5}$$
$$f'(0) = 0 \qquad (criticalpoint) \tag{6}$$
$$f''(x) = \frac{\partial(2x)}{\partial x} \tag{7}$$
$$f''(0) = 2 \qquad (notdegenerate) \tag{8}$$

$$Let f = x^3 \tag{9}$$
$$f'(x) = 3x^2 \tag{10}$$
$$f'(0) = 0 \qquad (criticalpoint) \tag{11}$$
$$f''(x) = \frac{\partial(3x^2)}{\partial x} = 6x \tag{12}$$
$$f''(0) = 0 \qquad (degenerate) \tag{13}$$

Generally, the behavior around degenerate critical points are hard to manage. So, we'll only consider functions such that no critical points are degenerate. This brings us (finally) to the definition of Morse functions.

Definition 1.4 (Morse Function) *A function $f : M \to \mathbb{R}$ is a* Morse function *iff the following conditions are met:*

1. *None of f's critical points are degenerate.*
2. *No two of f's critical points share the same function value.*

In other words, Morse functions are nicely behaving functions. Limiting our study only to Morse functions is not too restrictive as it turns out that the Morse functions form an open and dense subset of the space of all smooth functions $C^\infty(M)$ on M. So in this sense, a generic function is a Morse function. More importantly, considering such nice family of functions give us clean characterization of the topology induced by the function.

Lemma 1.5 (Morse Lemma) *Given a Morse function $f : M \to \mathbb{R}$, let p be a non-degenerate critical point of f, then there are local coordinate chart of a sufficiently small neighborhood of x such that (i) the coordinate of p is $(0, 0, 0)$ (the origin) in this chart, and (ii) locally the function f can be represented as*

$$f(x) = f(p) - x_1^2 - ...x_s^2 + x_{s+1}^2...x_d^2, \qquad for \quad s \in [0, d]$$

for every point $x = (x_1, x_2...x_d)$ in a small neighborhood of p.

The index *of a critical point p is s. There are $d+1$ types of critical points for a Morse function. Index-0 critical points are also called* minima*, while index-d critical points are also called* maxima*.*

Consider the example in Figure 3, where for a function f defined on a 2-manifold, there are only three types of critical points, with index-0 (minima), index-1 (saddle), and index-2 (maxima).

1. Local Minima: $f(x) = f(p) + x_1^2 + x_2^2$, when $s = 0$.
2. Saddle Point: $f(x) = f(p) - x_1^2 + x_2^2$, when $s = 1$.
3. Local Maxima: $f(x) = f(p) - x_1^2 - x_2^2$, when $s = 2$.

1.3 Connection to topology

First, a couple of definitions:

Definition 1.6 (Level Set) *Let $f : M \to \mathbb{R}$. Then all real numbers a have a preimage, $f^-1(a)$, known as a level set.*

$$M^a = f^-1(a) = \{x \in M \mid f(x) = a\}$$

Informally, a level set is a set of all the points in M that result in the same function value.

Definition 1.7 (Interval-Level Set) *Let $f : M \to \mathbb{R}$ and let $I \subseteq \mathbb{R}$. Then, the interval-level set is defined as:*

$$M^I = f^-1(I) = \{x \in M \mid f(x) \in I\}$$

An interval-level set is the union of all level sets for values from an interval I.

Definition 1.8 (Sublevel Set) *Let $f : M \to \mathbb{R}$ Then, the sublevel set is defined as:*

$$M^{(-\infty,a)} = \{x \in M \mid f(x) \leq a\}$$

And finally a sublevel set is a collection of all points with function values less than or equal to a certain value. It can be thought of as a sweeping through level sets.

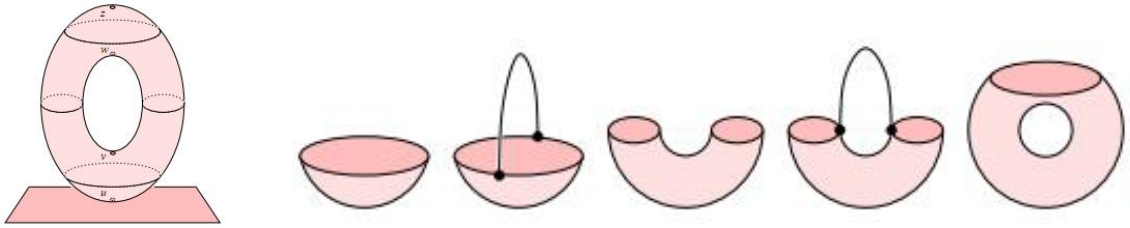
By examining the behavior of a sublevel set's topology as we sweep through a function, we can see that the following theorem is true:

Theorem 1.9 *$H_k(M^{(\infty,a]})$ is isomorphic to $H_k(M^{(\infty,b]})$ unless the interval $[a, b]$ contains a critical value.*

An important note to take away from this is that critical points denote where the topology of a sweeping of the sublevel set of a function will change.

We end this recap with an example; see Figure 4. Suppose we have a 2-manifold, in this case, a torus. And we define f to be the height function of each point on the torus as seen in Figure 4.

The level sets of this function would be the sets of points that share the same height or distance from the x-y plane. We can think of these sets as horizontal slices of the torus. And following this same scheme of thinking, a sublevel set of our height function would be sweeps of point from the base of the torus. A selected number of these sublevel sets are shown below in Figure 5.



(a) (b)

Figure 4: (a) The height function defined on a torus with critical points u, v, w, and z. (b) A few important sublevel sets (picture courtesy of [2]).

From this visualization, it is easy to convince ourselves of the truth of Theorem 2.4. As we sweep through the sublevel sets of the torus and cross through the critical points u, v, w, and z we can consider the Betti numbers of the sublevel set. At u β_0 increases to 1. β_1 is incremented at both v and w. And β_2 is incremented at z. These increments illustrate a fundamental change in topology.

2 PL setting

In the previous section, we consider the smooth case with smooth manifolds and smooth functions. However, often our domain is a piecewise-linear domain (i.e, a simplicial complex), and a natural family of functions defined on a simplicial complex is a piecewise-linear function.

Definition 2.1 (PL-functions) *Given a simplicial function K, and a function $f : V \to \mathbb{R}$ defined on vertices V of K, then f can be extended to a unique* piecewise-linear (PL) function $\tilde{f} : |K| \to \mathbb{R}$ *such that (i) $\tilde{f}|_V = f|_V$; and (ii) within each simplex $\sigma \in K$, $\tilde{f}$ is the linear.*

From now on, we will simplify notations and use f to denote $\tilde{f}$ as well. We will also abuse the notation slightly and talk about a PL-function $f : K \to \mathbb{R}$ (instead of $f : |K| \to \mathbb{R}$).

Stars and links. How do we talk about neighborhood in this simplicial complex setting? A *star* of a simplex τ is the set of simplices that have τ as a face, denoted by $\mathrm{St}\tau = \{\sigma \in K \mid \tau < \sigma\}$. Generally, the star is not closed under taking faces. We can make it into a complex by adding all missing faces. The result is the *closed star*, denoted by $\overline{\mathrm{St}}\tau$, which is the smallest subcomplex that contains the star. The $\mathrm{Lk}\tau$ consists of the set of simplices in the closed star that are disjoint from τ, $\mathrm{Lk}\tau = \{v \in \overline{\mathrm{St}}\tau \mid v \cap \tau = \emptyset\}$. Intuitively, we can think of the star of a vertex as a neighborhood around it, and the link as the boundary of that neighborhood.

The lower-star of a vertex v_i, denoted by $\mathrm{lowSt}(v_i)$, is the set of simplices whose vertices all have function values at most $f(v_i)$; that is, these simplices are spanned by vertices from $\{v_1, \ldots, v_i\}$. Similarly we can define the closed lower-star $\overline{\mathrm{lowSt}}(v_i)$ and lower-link $\mathrm{lowLk}(v_i)$.

Critical points. How can we define critical points in the PL-setting? Intuitively, by the Morse Lemma, the criticality of a point is completely determined by its lower-star / lower-link in the smooth case. In the smooth case assuming M is a Riemannian manifold with a Riemannian metric d_M defined on it, for any point $x \in M$, we can consider star of x being a small open neighborhood $\mathbb{B}_M(x, r) := \{y \in M \mid d_M(x, y) < r\}$ around x for a sufficiently small r. The link of x is the boundary of $\mathbb{B}_M(x, r)$. The lower star of x is the portion of star whose function value is at most $f(x)$. Similarly we can define the lower link.

Claim 2.2 (Classification of criticality for smooth case) *Given a smooth Morse function $f : M \to \mathbb{R}$ defined on a m-manifold M,*

1. *The star of any point $x \in M$ is homeomorphic to the open m-ball $\mathbb{B}^m$, and the link of x is homeomorphic to the $(m-1)$-sphere $\mathbb{S}^m$.*
2. *For a regular point x, its lower-star is homermorphic to the m-dimensional open half-space $\mathbb{R}^m_{>0}$, and its lower-link is homeomorphic to $\mathbb{B}^{m-1}$.*
3. *For an index-s critical point p, homology gropus of its lower star isomorphic to $\mathrm{H}_*(\mathbb{B}^s \times \mathbb{B}^{m-s})$, and the homology groups of the lower link are isomorphic to $\mathrm{H}_*(\mathbb{S}^{s-1} \times \mathbb{B}^{m-s})$.*

In the PL case, we will consider the *p-th reduced betti number* $\tilde{\beta}_p$ of the lower link of a vertex v: specifically, $\tilde{\beta}_{-1} = 0$, $\tilde{\beta}_0 = \beta_0 - 1$, and $\tilde{\beta}_i = \beta_i$ for $i > 0$.

Definition 2.3 (Classification of PL-critical points) *Given a PL function $f : K \to \mathbb{R}$, we say that a vertex $v \in K$ is* regular *if $\tilde{\beta}_i = 0$ for all $i \geq -1$; otherwise it is* critical. *A critical point v is simple if $\sum \tilde{\beta}_i = 1$.*

Furthermore, it is an index-k critical point if $\tilde{\beta}_k > 0$.

3 Persistence induced by functions

In the previous section, we assume that the input is a filtration of a simplicial complex K. Given a simplicial complex K, there are different ways to filter it. In this section, we consider a powerful framework to generate a filtration as induced by an input function f on K. This framework is powerful as we can now describe different properties of K depending on the specific function f; in particular, we can think of viewing K through the lens of f. The "features" that are persistent now corresponding to those with long "life-time" as measured by the function f.

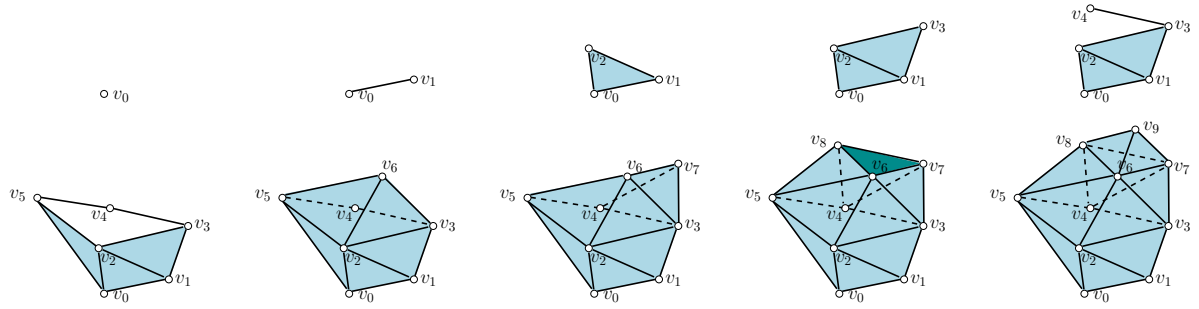


Figure 5: The function f is the height function. The sequence shows *lower-star* filtration of K induced by function f.

Lower-star filtration of K induced by f. Given a PL-function $f : K \to \mathbb{R}$, let $V = \{v_1, v_2, \ldots, v_n\}$ denote the set of vertices of K sorted, in a non-decreasing order, by their function values f. Define $K_i := \bigcup_{j \le i} \text{lowSt}(v_j)$ as union of the lower-star of all vertices whose f-function value is at most $f(v_i)$. Note that equivalently, K_i is the maximal subcomplex of K spanned by vertices in $V_i := \{v_1, \ldots, v_i\}$. The *lower-star filtration of K induced by f*, or simply *f-induced filtration*, is defined as the following filtration of K: (see Figure 5 for an example)

$$\mathcal{F}_f : \quad \emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_{n-1} \subseteq K_n = K. \tag{14}$$

Obviously, $\text{lowSt}(v_i) = K_i \setminus K_{i-1}$ for any $i \in [1, n]$.

Intuitively, imagine we sweep the domain K in increasing function values of f. At any point, we consider the *sublevel set* K^α, which is the set of points whose function value is below α. As α increases, this gives us a sequence of subsets of K, growing larger and larger. The above lower-start filtration is simply a discretization of this sequence where we include only discrete function values $\alpha_i = f(v_i)$ in this sequence.

How do we compute the persistence diagram for this filtration $\mathcal{F}_f$? Note that the algorithm that we introduced in Chapter **??** performs on a special type of filtration, where each time we only add one simplex to the current simplex. We call that type of filtration *simplex-wise filtration*. To compute the persistence homology, we first relax $\mathcal{F}_f$ into a simplex-wise filtration in the following way: Let $\mathcal{F}_s$ be any simplex-wise filtration induced by a total ordering of all m simplices in K

$$\sigma_1, \sigma_2, \ldots \sigma_{I_1}, \sigma_{I_1+1}, \ldots, \sigma_{I_2}, \sigma_{I_2+1}, \ldots, \sigma_{I_j}, \sigma_{I_j+1}, \ldots, \sigma_{I_{j+1}}, \ldots, \sigma_{I_{n-2}}, \ldots, \sigma_{I_{n-2}+2}, \ldots, \sigma_{I_{n-1}=m} \tag{15}$$

such that the following properties are satisfied:

(1) The total ordering induces a valid filtration; that is, for any simplex σ_i, the indices of its faces are smaller than i.

(2) Let $I_0 = 0$. We have that $\text{lowSt}(v_j) = \{\sigma_{I_{j-1}+1}, \ldots, \sigma_{I_j}\}$, for any $j \in [1, n]$.

The simplex-wise filtration induced by the above ordering is defined by

$$\mathcal{F}_s : \quad L_1 \subseteq L_2 \subseteq \cdots L_m, \quad \text{where} \quad L_i := \{\sigma_j \mid j \le i\} \text{ and thus } \sigma_i = L_i \setminus L_{i-1}. \tag{16}$$

In other words, Condition (1) above enforce that L_i is a valid simplicial complex for any i. Condition (2) means that $K_i = L_{I_i}$; thus $\mathcal{F}_s$ is a realization of $\mathcal{F}_f$ in simplex-wise filtration format.

The choice of $\mathcal{F}_s$ is not unique. We can simply obtain $\mathcal{F}_s$ by set $\sigma_{I_{j-1}+1}, \ldots, \sigma_{I_j}$ as the set of simplices in lowSt(v_j) sorted by their dimension (which would then guarantee Condition (1) above). We now set up the map $\pi : [0, m] \to [0, n]$ as $\pi(j) = k$ if $j \in [I_{k-1} + 1, I_k]$; that is, $\pi(j)$ means that simplex σ_j is in the lower-star of vertex v_k.

Finally, we perform the persistence algorithm on the filtration $\mathcal{F}_f$. However, we set up the persistence diagram w.r.t the filtration $\mathcal{F}_f$, denoted by $\mathcal{D}f$, as follows:

Definition 3.1 (Persistence pairing of critical points) *If there is a persistence pairing $(i, j) \in \mathcal{D}\mathcal{F}_s$ (i.e, $\mu i, j \geq 1$ w.r.t. $\mathcal{F}_s$), then we say that there is a* pairing of (critical) vertices $(v_{\pi(i)}, v_{\pi(j)})$ *if and only $\pi(i) \neq \pi(j)$.*

This gives rise a persistent point $(f(v_{\pi(i)}), f(v_{\pi(j)}))$ *in the persistence diagram $\mathcal{D}f$, with persistence being $|f(v_{\pi(i)}) - f(v_{\pi(j)})|$.*

A simple 1D example is given in Figure **??**.

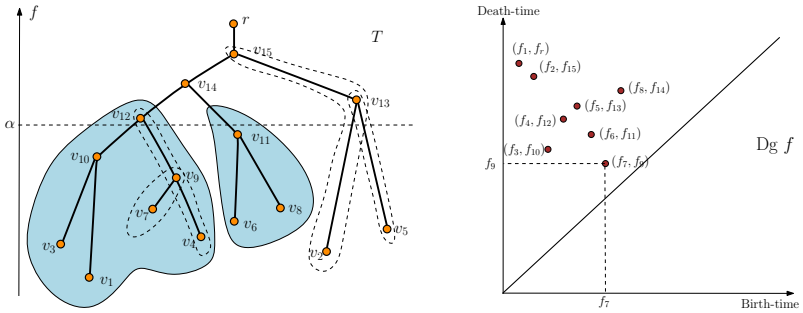


Figure 6: (a) We plot the 1-dimensional simplicial complex K so that the height of a point is its f value. The sublevel set K_α is the portion of T lying below the horizontal dashed line. Consider the merge at branching point $s = v_{14}$: two components merged at v_{14} (shaded regions) are generated by minimum $m_1 = v_1$ and $m_2 = v_6$ respectively. This gives rise to a persistence point $(f(v_6), f(v_{14}) = (f_6, f_{14})$ in the persistence diagram in (b). In (b), for simplicity, we set $f_i := f(v_i)$. We mark some pairs of tree nodes generating persistence points in (a) via dashed closed curves, such as (v_7, v_9) and (v_4, v_{12}).

While they are not immediately evident, the following two results hold:

Lemma 3.2 *The set of vertices that contribute the persistence pairings in $\mathcal{D}f$ are all PL-critical. Furthermore, a pairing is always between two critical points whose indices differ by* 1.

Lemma 3.3 *Any simplex-wise filtration $\mathcal{F}_s$ induced by the low-star filtration $\mathcal{F}_f$ will give rise the the same pairing between critical points and the same persistence diagram.*

Finally, we remark that, instead of sublevel sets filtration, we can also sweep the domain top-down and consider the upper-star filtration induced by a function. This typically captures different information about f w.r.t f. See Figure 7 for a simple 1D example.

4 Stability of Persistence diagrams

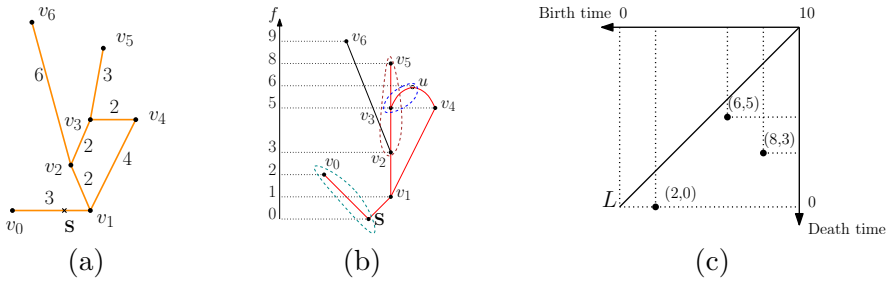


Figure 7: (a) A graph, (b) with a function f defined on it, shown as the height function; and (c) its corresponding persistence diagram for upper-star f-induced filtraiton.

Metric on persistence diagrams. Given two finite persistence diagrams $\mathrm{Dg} = \{p_1, \ldots, p_\ell \in \mathbb{R}^2\}$ and $\mathrm{Dg}' = \{q_1, \ldots, q_k \in \mathbb{R}^2\}$, a common distance measure for them, the *bottleneck distance* $d_B(\mathrm{Dg}, \mathrm{Dg}')$ [**?**], is defined as follows: Consider Dg and Dg′ as two finite sets of points in the plane (where points may overlap). Call $L = \{(x, x) \in \mathbb{R}^2\}$ the *diagonal* of the birth-death plane.

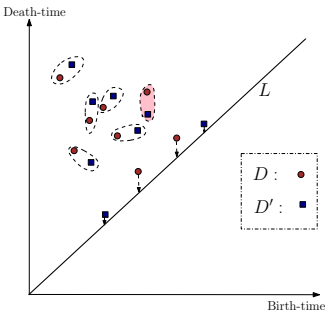


Figure 8: Bottleneck matching between two persistence diagrams D and D' (red and blue squares, respectively).

Definition 4.1 *A* partial matching C *of* Dg *and* Dg′ *is a relation* $C : (\mathrm{Dg} \cup L) \times (\mathrm{Dg}' \cup L)$ *such that each point in* Dg *is either matched to a unique point in* Dg′*, or mapped to its closest point (under* L_∞*-norm) in the diagonal* L*; and the same holds for points in* Dg′*. See Figure 8. The bottleneck distance is defined as* $d_B(\mathrm{Dg}, \mathrm{Dg}') = \min_C \max_{(p,q)\in C} \|p - q\|_\infty$*, where* C *ranges over all possible partial matchings of* Dg *and* Dg′*. We call the parital matching that achieves the bottleneck distance* $d_B(\mathrm{Dg}, \mathrm{Dg}')$ *as the* bottleneck matching*.*

The bottleneck distance is sensitive to outliers. One can change the *max* distance between corresponding points to the sum of distances between all corresponding pairs of points. This gives rise to the L_1-Wasserstein distance between two persistence diagrams [2]. One can further convert the persistence diagram to the so-called *persistence landscapes* introduced by Bubenik [1], and measure their distance as the norm in a functional space.

Time varying functions. Now imagine we have a *time-varying function* defined on X, which is a continuous function $F : X \times \mathbb{R} \to \mathbb{R}$ such that $F(x, t)$ gives the function value of $x \in X$ at time t. Let $F_t : X \to \mathbb{R}$ denote the function at time t, defined as $F_t(\cdot) = F(\cdot, t)$. This continuous function F_t defined on X varies continuously as time $t \in R$ changes. It turns out that the persistence diagram of this time varying function can be summarized in a structure that we can a *vineyard*.

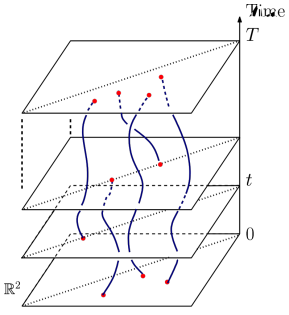


Figure 9: The vertical axis is time, and horizontal $\mathbb{R}^2$ contains persistence diagrams.

In particular, consider the function F_t at time t. Let $\mathcal{D}F_t$ denote the persistence diagram at this time. As t varies continuously to t', by the Stability Theorem, $\mathcal{D}F_t$ changes continuously to $\mathcal{D}F_{t'}$ since F_t changes continuously. In particular, each persistence point in $\mathcal{D}F_t$ traces out a curve, called a "vine". The collection of such

vines form the "vineyard", representing the persistence summary for the time varying function F as a whole. See the right figure for an example. Note that a vine can be created, and terminates at certain times. When it happens, it is necessarily at a diagonal point.

5 Persistence algorithm for 0-th persistent homology

While the general persistence algorithm runs in $O(N^3)$ time, where N is the total number of simplicies in the filtration, it turns out that for the 0-th persistent homology (0-th persistence diagram) for a PL function $f : K \to \mathbb{R}$ can be computed efficiently in $O(n \log n + m\alpha(n))$ time, where n and m are the number of vertices and edges in K, respectively.

Indeed, first, observe that to compute $\mathcal{D}_0 f$, we only need the 1-skeleton of K, that is, vertices V and edges E. Assume that all vertices in V are sorted in non-decreasing order of their f function values. Let K_i be the union of lower-stars of all v_j, with $j \leq i$. Since we are only interested in the 0-th homology, we only need to track the 0th homology groups in K_i, which is the connected components information.

Assume we are at vertex v_j. Consider $\text{lowSt}(v_j)$. There are three cases.

Case-1 : $\text{lowSt}(v_j) = \{v_j\}$. Then v_j starts a new connected component in K_j. Hence v_j is a creator.

Case-2 : All edges in $\text{lowSt}(v_j)$ connect to vertices from the same connected component C in K_{j-1}. In this case, the component C grows, in the sense that it now include also vertex v_j (and its incident edges in its lower-star). However, the homology group of K_i are isomorphic to that of K_{i-1}.

Case-3 : Edges in $\text{lowSt}(v_j)$ link to two or more components $C_1, \ldots, C_a$ in K_{j-1}. In this case, after the addition of $\text{lowSt}(v_j)$, all $C_1, \ldots, C_a$ are merged into a single component

$$C' = \Big(\bigcup_{i \in [1,a]} C_i \Big) \bigcup \text{lowSt} v_j. \tag{17}$$

Hence inclusion $K_{j-1} \hookrightarrow K_j$ induces a surjective homomorphism $\xi : \text{H}_0(K_{j-1}) \to \text{H}_0(K_j)$ and $\beta_0 - = (a-1)$. That is, we can consider that $a-1$ number of components are killed, only one stays on as C'.

It turns out that when Case-3 happens, we can imagine that the following is happening: For each component C in K_{j-1}, its corresponding homology class $[C]$ has a preimage in any K_i with $i \geq i_c$, where i_c is the index of the global minimum in C. However, it does not have image in K_i for $i < i_c$. In other words, intuitively, C was first created in K_{i_c} when we process the vertex v_{i_c}, which is the global minimum in this component C. Or in a simplified manner, we say that C is *created* at its global minimum v_c. After all C_i's, $i \in [1, a]$ merge into C' as in Eqn (17), the new component C' is created at its global minimum. In other words, assuming that C_1 has the lowest global minimum v_{c_1} than all other C_i with $i \in [2, a]$, then C' is created at v_{c_1}; that is, the homology class $[C_1]$ survives. (More precisely, we should think that the homology classes $[C_1 + C_i]$ for each $i \in [2, a]$, was created at the global minimum of C_i, and killed at time j.) We thus obtain a set of pairing of critical points (v_{c_i}, v_j), where v_{c_i} is the global minimum of component C_i, for $i \in [2, a]$.

Finally, to develop an algorithm to achieve what we describe above, note that we only need to maintain connected components information for each K_i, and potentially merge multiple components. We would also need to be able to query for which component in the current sublevel sets contain a given vertex u. Such operations can be implemented by a standard union-find data structure. We provide the algorithm $\mathsf{Comp0thDg}(K)$ shown in Algorithm 5; note that in this algorithm,

Algorithm 1 Comp0thDg($K = (V, E)$)

```
Sort all vetrtices in V;
for j = 1 → n do
  CreateSet(v_j);
  Set flat = 0;
  for any (v_k, v_j) ∈ lowSt(v_j) do
    if (flag == 0) then
      Union(v_k, v_j);
    else
      if FindSet(v_k) ≠ FindSet(v_j) then
        Union(v_k, v_j);
        Output pairing (RepSet(v_k), RepSet(v_j));
      end if
    end if
  end for
end for
```

we assume that we use the minimum of a set as its *representative*, and the query RepSet(v) returns the representative of the set containing vertex v. We assume that this query takes the same time as FindSet(v).

To analyze the time complexity, let n and m denote the number of vertices and edges respectively. Note that sorting takes $O(n \log n)$ time. There are then $O(n + m)$ number of CreateSet, FindSet, Union and RepSet operations. By using the standard union-find data structure, the total time for all these operations are $(n + m)\alpha(n)$. Hence the total time complexity of Algorithm Comp0thDg(K) is $O(n \log n + m\alpha(n))$.

References

[1] P. Bubenik. Statistical topology using persistence landscapes. *CoRR*, abs/1207.6437, 2012.

[2] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., Providence, Rhode Island, 2010.